

# Assignment #8

Jingye Wang

November 3, 2016

```
setwd("~/Dropbox/WUSTL third/Multilevel Modeling for Quantitative Research/assignment/8")
.libPaths("/Library/Frameworks/R.framework/Versions/3.3/Resources/library")
```

## Ch18.1

Linear regression algebra: show that weighted least squares is maximum likelihood estimation for the model (18.7).

Since  $y_i \sim N(X_i\beta, \sigma^2/w_i)$  and  $N(y_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}(\frac{y_i - \mu}{\sigma})^2)$ ,

we want to maximize  $N(y_i | X_i\beta, \frac{\sigma^2}{w_i}) = \frac{1}{\sqrt{2\pi} \frac{\sigma}{\sqrt{w_i}}} \exp(-\frac{1}{2}(\frac{y_i - X_i\hat{\beta}}{\sigma/\sqrt{w_i}})^2)$ ,

which is the same to minimize  $\sum_{i=1}^n (\frac{y_i - X_i\hat{\beta}}{1/\sqrt{w_i}})^2 \Rightarrow \sum_{i=1}^n w_i (y_i - X_i\hat{\beta})^2$ .

$\sum_{i=1}^n w_i (y_i - X_i\hat{\beta})^2$  is the weighted sum of squares. So, weighted least squares is maximum likelihood estimation for the model.

## Ch18.2

Bayesian inference: take a multilevel linear model that you have already fit, and make a graph such as in Figure 18.5 or 18.6 showing likelihood, prior distribution, and posterior distribution, in each of several groups.

I'm going to use the model in Ch12.2, Part A.

```
library(lme4)
library(arm)
library(rjags)
library(R2jags)
library(R2WinBUGS)

## Import dataset
hiv_data <- read.csv ("allvar.csv", header= T)
hiv_data$time <- hiv_data$visage - hiv_data$baseage
CD4PCT <- hiv_data$CD4PCT
n <- length(CD4PCT)
y <- sqrt(CD4PCT)
x <- hiv_data$time
id.name <- as.vector(hiv_data$newpid)
uniq <- unique(id.name)
J <- length(uniq)

## Complete pooling regression
lm.pooled <- lm (y ~ x, hiv_data)
display (lm.pooled)

## No pooling regression
lm.unpooled <- lm (y ~ x + factor(newpid)-1, hiv_data)
display (lm.unpooled)

## Multilevel model with no group-level predictors
M1 <- lmer (y ~ x + (1 | newpid), hiv_data)
display (M1)

num <- c (2, 11, 14, 16) # counties to display

## Plot
coef.unpooled <- coef(lm.unpooled)
se.unpooled <- se.coef(lm.unpooled)

mu.a <- fixef(M1)[1]
sigma.a <- 0.33
a.j <- rep(NA,J)
a.j <- fixef(M1)[1] + ranef(M1)$newpid
a.j.sigma <- rep(NA,J)
a.j.sigma <- se.coef(M1)$newpid

par(mfrow=c(3,4))
for (j in num){
  curve(dnorm(x, coef.unpooled[j+1], se.unpooled[j+1]), from=-0.5, to=8.5,
    xlab=expression(alpha[j]), ylab="", main=paste(uniq[j],":likelihood"),
    yaxt="n", bty="n", cex.lab=1.2, cex.axis=1.1, cex.main=0.8)
}

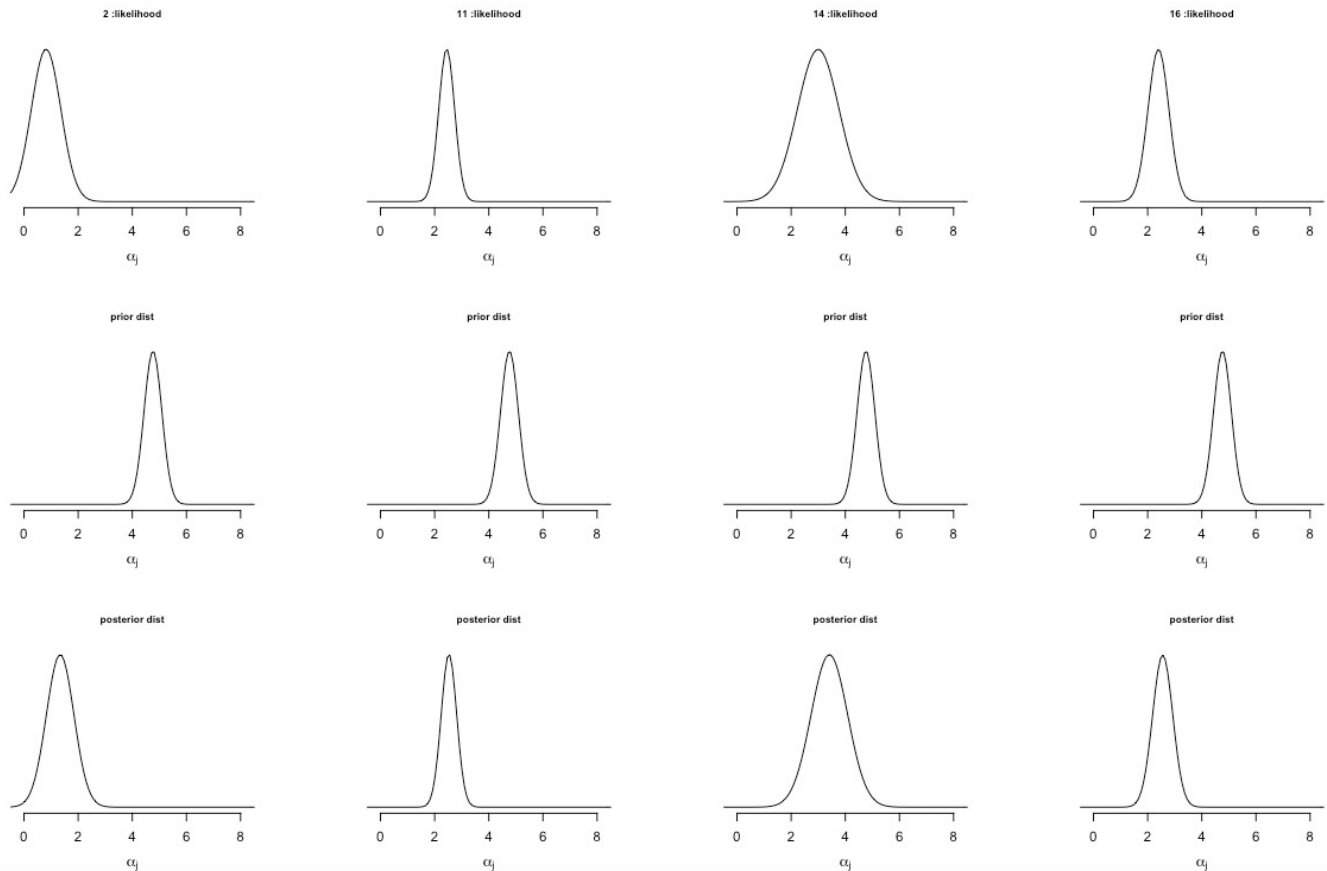
for (j in num){
  curve(dnorm(x, mu.a, sigma.a), from=-0.5, to=8.5,
    xlab=expression(alpha[j]), ylab="", main=paste("prior dist"),
    yaxt="n", bty="n", cex.lab=1.2, cex.axis=1.1, cex.main=0.8)
```

```

}

for (j in num){
  curve(dnorm(x, a.j[j,], a.j.sigma[j]), from=-0.5, to=8.5,
    xlab=expression(alpha[j]), ylab="", main=paste("posterior dist"),
    yaxt="n", bty="n", cex.lab=1.2, cex.axis=1.1, cex.main=0.8)
}

```



## Ch18.4

Censored data: take the data on beauty and teaching evaluations data described in Exercise 3.5 and artificially censor by reporting all course evaluations below 3.0 simply as “\*\*”

```

## Import dataset
beauty_data <- read.csv ("ProfEvaltnsBeautyPublic.csv", header= T)

```

## Part A

Take one of the models from that earlier exercise and write the likelihood function given this mix of observed and censored data.

$$y_i = \begin{cases} z_i & \text{if } z_i \geq 3.0 \\ 3.0^- & \text{if } z_i < 3.0 \end{cases}$$

$$P_r(y = 3.0^-) = P_r(z_i < 3.0) = \int_0^{3.0} N(z_i | a + bx_i, \sigma^2) = \Phi\left(\frac{a + bx_i - z_i}{\sigma}\right)$$

$$P_r(y = z_i) = P_r(z_i \geq 3.0) = N(y_i | a + bx_i, \sigma^2)$$

## Part B

Find the maximum likelihood estimate in R using the `optim()` function.

```
## Rename the two variables for convenience
beauty <- beauty_data$btystdave
evaluation_raw <- beauty_data$courseevaluation
evaluation <- evaluation_raw
C <- 3.0
evaluation[evaluation < C] <- C

## Maximum likelihood estimate
Loglik <- function (parameter.vector, x, y, C) {
  a <- parameter.vector[1]
  b <- parameter.vector[2]
  sigma <- parameter.vector[3]
  ll.vec <- ifelse (y > C, dnorm (y, a + b*x, sigma, log=TRUE),
                  pnorm ((a + b*x + C)/sigma, log=TRUE))
  return (sum (ll.vec))
}
inits <- runif (3)

## run optim function
mle <- optim (inits, Loglik, lower=c(-Inf, -Inf, 1.e-5), method="L-BFGS-B", control=list
(fnscale=-1), x=beauty, y=evaluation, C=3)
mle$par
```

```
> mle$par
[1] 4.08230627 0.09649986 0.46736776
```

## Part C

Fit the model using Bugs, accounting for the censoring.

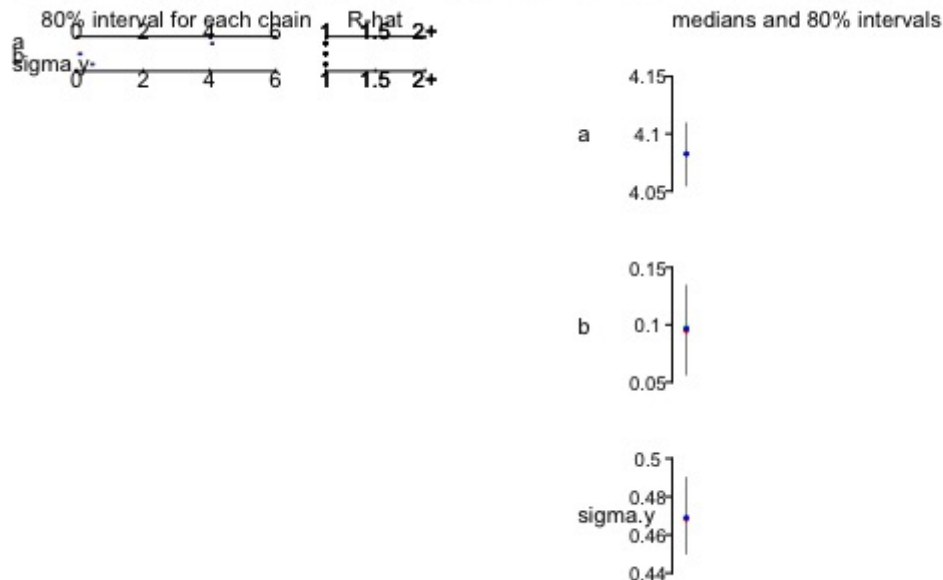
```
## Fitting the censored-data model using Bugs
model_censored <- function() {
  for (i in 1:n){
    z.lo[i] <- dinterval(y[i], C)
    z[i] ~ dnorm (z.hat[i], tau.y)
    z.hat[i] <- a + b*x[i]
  }
  a ~ dnorm (0, .0001)
  b ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)
}

z <- ifelse (evaluation==C, NA, evaluation)
n <- length(beauty)
data <- list (x=beauty, y=evaluation_raw, z=z, n=n, C=C)
inits <- function() {
  list (a=rnorm(1), b=rnorm(1), sigma.y=runif(1))
}
params <- c ("a", "b", "sigma.y")
censoring.1 <- jags (data, inits, params, model_censored, n.chains = 3, n.iter=1000, DIC
=F)
censoring.1 %>% plot
censoring.1$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//Rtmp8XPQY
J/model108a1508e405.txt", fit using jags,
 3 chains, each with 1000 iterations (first 500 discarded)
n.sims = 1500 iterations saved
      mean sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
a       4.1  0   4.0 4.1 4.1 4.1   4.1    1  1500
b       0.1  0   0.0 0.1 0.1 0.1   0.2    1  1500
sigma.y 0.5  0   0.4 0.5 0.5 0.5   0.5    1  1500
```

For each parameter, n.eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

12rb3n3gs4zx9\_3\_xly5fw0000gn/T//Rtmp8XPQYJ/model108a1508e405.txt", fit using jags, 3 chains, each with 10



## Part D

Compare the censored-data inferences from the estimates using the complete data.

```
lm_complete<- lm(evaluation_raw ~ beauty)
display(lm_complete)
```

```
lm(formula = evaluation_raw ~ beauty)
      coef.est coef.se
(Intercept)  4.01    0.03
beauty        0.13    0.03
---
n = 463, k = 2
residual sd = 0.55, R-Squared = 0.04
```

In the JAGS model,  $a=4.1$ ,  $b=0.1$  ( $y=0.1x+4.1$ ); while in the complete model,  $a=4.01$ ,  $b=0.13$  ( $y=0.13x+4.01$ ). We can see that the slope of JAGS model is smaller than the complete model and the intercept of of JAGS model is larger than the complete model.